



ITW
AT
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE JACOBS et al.

Application for Patent

Filed June 22, 2000

Serial No. 09/602,576

FOR:

INTEGRATED PLATFORM FOR DEVELOPING AND MAINTAINING A
DISTRIBUTED MULTIAPPLICATION ONLINE PRESENCE

APPEAL BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as first-class mail on October 28, 2004 in an envelope addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450.

Signed:

Deborah Neill

11/01/2004 KBETEM1 00000113 09602576

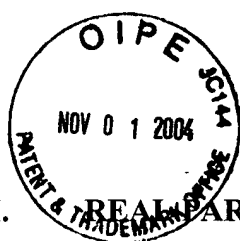
02 FC:2402

170.00 OP

BEYER WEAVER & THOMAS, LLP
Attorneys for Appellants

TABLE OF CONTENTS

	<u>Page No.</u>
I. REAL PARTY IN INTEREST	1
II. RELATED APPEALS AND INTERFERENCES	1
III. STATUS OF CLAIMS	1
IV. STATUS OF AMENDMENTS	1
V. SUMMARY OF CLAIMED SUBJECT MATTER	1
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL	4
VII. ARGUMENT	4
A. The Homestead Reference Fails to Support a Rejection of Claims 1-17 and 19-30 Under 35 U.S.C. § 102(a).	5
1. The Homestead Reference Fails to Disclose or Suggest A Data Schema Having an Underlying Extensible Data Model, as Required to Anticipate Claims 1-17 and 19-30.	6
2. The Homestead Reference Fails to Disclose or Suggest Panel Objects Through Which Data are Entered and Manipulated, as Required to Anticipate Claims 24 and 25.	10
B. The Homestead Reference Fails to Support a Rejection of Claim 31 Under 35 U.S.C. § 103(a).	12
C. The Homestead and Norcott References Fail to Support a Rejection of Claim 18 Under 35 U.S.C. § 103(a).	13
D. Conclusion	14
VIII. CLAIMS APPENDIX	16



I. REAL PARTY IN INTEREST

The real party in interest is Affinity Internet, Inc., 3250 West Commercial Blvd., Fort Lauderdale, Florida 33309, dba Bigstep.com.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals, interferences, or judicial proceedings known to the Appellants.

III. STATUS OF CLAIMS

There are a total of 31 claims pending in this application (claims 1-31), all of which were submitted with the application as filed. Claims 1-17 and 19-30 were rejected under 35 U.S.C. § 102(a) as anticipated by Homestead.com ("Homestead"). Claim 31 was rejected under 35 U.S.C. § 103(a) as obvious in view of Homestead. Claim 18 was rejected under 35 U.S.C. § 103(a) as obvious in view of Homestead and Norcott et al., U.S. Patent No. 6,334,128 ("Norcott"). Appellants appeal all of the rejections of claims 1-31 and respectfully request reversal of the rejections.

IV. STATUS OF AMENDMENTS

No amendments were filed following the second and final Office Action of April 6, 2004 ("Office Action").

V. SUMMARY OF CLAIMED SUBJECT MATTER

Aspects of the present invention generally relate to software architectures, platforms, and data constructs that provide a system for enabling a non-technical or lay user to perform

discrete technical tasks necessary to build a complete network-based, multiuser application. The system also allows the user to have a uniform user experience throughout development of the application. For example, the platform can be used to construct and maintain an Internet or online Web site capable of handling e-commerce transactions or can be used to develop a customer relationship management system. (Application as filed, page 11, line 10 – page 12, line 13).

In one aspect of the invention, as illustrated in FIG. 2, a software architecture is described that enables multiple users to perform a variety of tasks via a wide-area network, such as an enterprise network or the Internet. (Application as filed, page 14, line 16 – page 15, line 8). The architecture has an extensible data schema 102 for storing data objects. The data schema 102 has an underlying data model, described below, that can be extended with previously undefined attributes without having to alter the basic format of the schema 102. In FIG. 1, the architecture also includes an integrated platform 100 that enables each of the users to perform the tasks by controlling interaction or communication between the services and systems, and the extensible database. (Application as filed, page 12, line 14 – page 14, line 15). The architecture and platform allow each user and application to extend the database in a user-specific way. This allows users and applications to use the same database, without having to change the underlying structure of the database. (Application as filed, page 6, line 23 – page 7, line 3).

In another aspect of the present invention, an integrated software platform for creating an end-user application having a uniform user experience and a single-registration feature is described. As shown in FIGs. 3 and 4, a data model 300 arranges and configures application data 400, which are either of a fixed attribute type or an extended attribute type. (Application as filed, page 19, lines 12-22). The platform also includes a data logic or dependency logic

component 410 for operating on the application data. A back-end code layer 104, shown in FIG. 1, manages the uniform user experience and a visual design component implements the user experience primarily by presenting a user interface that is used for entering data into the system. The user interface can be uniform across the multiple applications used “behind the scenes” to build the network-based application desired by the user. The platform communicates with a data schema or repository for storing the application data configured in the data model 300.

In another aspect of the present invention, as illustrated in FIG. 2, a task-based architecture for building a multiuser, online application through a task-based approach is described. (Application as filed, page 14, line 16 – page 15, line 8). The architecture includes a data schema 102 for storing data related to the online application. A data model 202 stores and shares the data as multiple tasks are completed and is physically stored in the data schema 102. The task-based architecture also includes multiple tools 206 and a task viewer application 208 for creating a user interface. Several services allow a user to gather and author the data.

In another aspect of the present invention, illustrated in FIGs. 8 and 9A-9C, a system for building a distributed, multiapplication program is described. The system includes multiple tasks, where a task is made up of one or more sequences 502, as shown in FIG. 5. The system also includes multiple panel objects 506 within a sequence through which data related to the multiapplication program is entered and manipulated. A panel object is aware of a data object to access to retrieve existing application data. (Application as filed, page 24, lines 3-17).

In yet another aspect of the present invention, illustrated in FIGs. 7, 9A, and 10, a method of building a customized Web site is described. One or more Web pages, such as a

home page or a dynamic map page is created and maintained. (Application as filed, page 36, line 1 – page 37, line 9). A communication service 704 for communicating with users accessing the Web site is developed. (Application as filed, page 25, line 22 – page 26, line 4). An online transaction system for processing online orders made through the Web site is also developed. A reporting service 712 for generating reports relating to Web site activity is created. (Application as filed, page 26, lines 15-17). The method of building the customized Web site involves a task-based approach to completing an activity and has a uniform user experience.

VI. GOUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The rejections presented for review are as follows:

- A. Claims 1-17 and 19-30 were rejected under 35 U.S.C. § 102(a) as anticipated by Homestead.
- B. Claim 31 was rejected under 35 U.S.C. § 103(a) as obvious in view of Homestead.
- C. Claim 18 was rejected under 35 U.S.C. § 103(a) as obvious in view of Homestead and Norcott.

VII. ARGUMENT

With respect to Ground “A” above, the rejected claims do not stand or fall together. Claims 1-17, 19-23 and 26-31 are presented as a first group, and claims 24 and 25 are presented as a second group. The second group of claims is separately patentable over claim 1 for the reasons below.

With respect to Ground “B,” no grouping is possible since only claim 31 was rejected.

With respect to Ground “C,” no grouping is possible since only claim 18 was rejected.

A. The Homestead Reference Fails to Support a Rejection of Claims 1-17 and 19-30 Under 35 U.S.C. § 102(a).

Claims 1-17 and 19-30 were rejected under 35 U.S.C. § 102(a) as anticipated by a combination of archived web materials from Homestead.com (“Homestead”). Homestead describes a web site which enables users to build their own web sites without programming. (Homestead, page 1). The Homestead site provides web page templates and “drag and drop” elements with which users can construct their sites. (Homestead, page 8).

Homestead is largely irrelevant to the invention as claimed, apart from the superficial similarity with one use of the present invention to build a web site. Many of the recited features throughout the claims of the present application are neither described nor even remotely suggested in Homestead. Appellants respectfully submit that this rejection should be reversed.

Claim 1, by way of example, defines a software architecture for enabling multiple users to perform a plurality of tasks. The software architecture includes:

a plurality of applications;

a data schema for storing a plurality of data objects, the data schema having an underlying, extensible data model; and

an integrated platform for enabling each of the multiple users to perform at least one of the tasks by controlling interaction between two or more of the applications and the extensible data model.

As explained below, Homestead fails to disclose or reasonably suggest the features of a software architecture such as that defined by claim 1.

1. The Homestead Reference Fails to Disclose or Suggest A Data Schema Having an Underlying Extensible Data Model, as Required to Anticipate Claims 1-17 and 19-30.

Claim 1, by way of example, defines a software architecture comprising:

a data schema for storing a plurality of data objects, the
data schema having an underlying, extensible data model.
(Emphasis added.)

As explained in the application as filed, the claimed software architecture enables many users to perform a variety of tasks via a wide-area network, such as an enterprise network or the Internet. In one embodiment, the architecture has several services, systems, and an extensible database for storing data objects. The database has an underlying structure referred to as a “**data schema**” that can be extended with previously undefined attributes without having to alter the basic format of the schema. (Application as filed, page 6, lines 12-20). As described with reference to FIG. 4, in one embodiment, the data schema physically stores the data used by the applications, tools, and functions in conjunction with a task-based architecture. The data schema can store data in the form of a relational database, flat files, a multidimensional database, or any other appropriate data storage schema. (Application as filed, page 13, lines 3-6).

A “**data model**,” described in one embodiment with reference to FIGS. 3 and 4, represents an arrangement or configuration of data in terms of fixed and extensible attributes as well as logic for operating on the data. (Application as filed, page 13, lines 19-22). Data arranged in the data model can be, for example, business data and data relating to users, where a user is an SBO or online merchant. (Application as filed, page 13, lines 22-24). The data

model has a unique configuration that facilitates the collection and aggregation of data and the addition of previously undefined data attributes, referred to as “**extensibility**.” (Application as filed, page 12, line 24 – page 13, line 2). By extending the attributes of a data model, the platform can process the disparity among different types of data. For example, a data object can expose its data to a model object via an XML layer (shown as XML layer 203 of FIG. 2), thereby allowing for attributes to be added as desired, *i.e.*, making them extensible. Model objects can be extended as new relationships form which require enforcement of data dependencies. (Application as filed, page 22, lines 16-20).

In one example, as described in the Application as filed at page 22, line 21 through page 23, line 14, a model object acts on a relationship between a small business owner (“SBO”) and an online customer. The model object enforces a dependency binding a data object to a business entity. Extensible attributes of such a data object may begin with data relating to when the relationship was established, how the customer was referred to the SBO online site, and whether a transaction was completed. Using XML to define these attributes facilitates extending the SBO’s Web site, to be aware of the number of return visits by the customer without requiring the SBO to have an understanding of or having to modify the underlying storage structures or dependencies. Over time, the XML definition for such a “customer relationship” object may expand to cover all aspects of the history of transactions between the business and the customer. By virtue of XML, the original specification describing when the customer first visited the Web site remains unchanged while whole new subcategories of information have been added to the object. The overall model has added no new rules to support the new data, nor has the underlying data layer and storage system changed to support the potentially vast data contained by the system.

The Office Action asserts on page 2, paragraph 4, that “Homestead discloses . . . a data schema for storing a plurality of data objects, the data schema having an underlying, extensible data model (at least pp. 41, 49-52).” The Examiner is apparently attempting to equate: (1) a web page customized with predefined applets or “Elements,” with (2) a data schema having an underlying, extensible data model. Homestead simply does not support this assertion, as there are no similarities on which to base the comparison.

The “Elements” disclosed in Homestead are a collection of predefined applets. Homestead describes these Elements as “the basic building blocks of web sites.” (Homestead, page 7). Because the Homestead Elements are predefined, they can be stored in Homestead’s “Element palette.” (Homestead, page 41). This palette, or collection of predefined applets, is presented to a user. The user can then select desired Elements and “drag-and-drop” them onto a web page. (Homestead, page 51). The Homestead Elements can be functional applets, such as search engines, hit counters, web polls, news tickers, and animated text. However, Homestead fails to disclose or suggest any data schema having an underlying, extensible data model to support or store these applets. This is probably because all of the Homestead “Elements” are predefined. While the Homestead Elements or applets can be used in a dynamic manner, such as a photo album, guestbook, or chatroom (Homestead, page 8), the applets themselves are predefined. Nowhere in the Homestead reference are any of the various Elements described as “extensible” or part of an “extensible data model.” The Examiner made this characterization in hindsight.

Claim 1 recites “a data schema for storing a plurality of data objects, the data schema having an underlying, extensible data model.” Accordingly, a data schema defined according to claim 1 can be extended with previously undefined attributes without having to alter the basic format of the schema, as explained above. Homestead fails to disclose or suggest any

data schema, much less a data schema having an underlying, extensible data model. This is not surprising given that the Homestead reference is promotional in nature and not targeted to an audience which would have any appreciation or interest in such information. By contrast, the “underlying, extensible data model” feature of claim 1 provides for the aggregation of data and addition of previously undefined data attributes. Thus, for example, in the SBO-customer model described above, a definition for an entire relationship between the SBO and customer can be established and then expanded to cover all aspects of the transactions between the business and customer. By incorporating the underlying, extensible data model of the data schema, the relationship is defined and expanded without having to add new rules to support new data, or having to change the structure of the underlying data model.

The “extensibility” feature and associated benefits of claim 1 described above are not provided by Homestead, because Homestead fails to disclose or suggest a “data schema having an underlying, extensible data model” in the first place. An “underlying, extensible data model” of a data schema cannot be implied by a web site which simply allows dragging and dropping hit counters, web polls, and guestbooks onto a personalized web page. No such inference is warranted. Any of a variety of conventional software architectures could be employed to construct the Homestead site and to provide the functionality described. And because no details are provided regarding the nature of the software architecture or the associated data model employed, a conventional architecture and data model should be assumed. The Examiner is not entitled to use hindsight to propose that the novel and nonobvious software architecture described and claimed in the present application underlies the Homestead site.

Other references to the Homestead reference in the Office Action, such as the rejection of claim 13 based on pages 2-3, 7-8 and 40 are also misplaced. None of the cited pages

include any information describing a data model underlying the Homestead site. The simple fact of the matter is that there are no references to an underlying data model anywhere in the Homestead materials. Because Homestead fails to disclose or suggest any data schema having an underlying, extensible data model, it was improper to reject the claims as anticipated by Homestead under 35 U.S.C. § 102(a). This rejection should be reversed.

Claims 2-12 are dependent upon claim 1 and are, therefore, not subject to rejection under 35 U.S.C. § 102(a) for at least the same reasons as claim 1. Independent claims 13, 21, 24 and 26 recite similar features as claim 1 and are, therefore, not subject to the anticipation rejection for similar reasons as claim 1. Claims 14-17, 19 and 20 are dependent upon claim 13, claims 22 and 23 are dependent upon claim 21, claim 25 is dependent upon claim 24, and claims 27-30 are dependent upon claim 26. These dependent claims are, therefore, not subject to rejection under 35 U.S.C. § 102(a) for at least the same reasons as the independent claims on which they are based.

2. The Homestead Reference Fails to Disclose or Suggest Panel Objects Through Which Data are Entered and Manipulated, as Required to Anticipate Claims 24 and 25.

Claim 24 defines a system for building a distributed, multi-application program, in which:

one or more panel objects within a sequence through which data related to the multi-application program is entered and manipulated, wherein a panel object is aware of which data object from the plurality of data objects to access to retrieve existing data related to the multi-application program. (Emphasis added.)

As explained in the Application as filed, page 24, lines 4-13, it is at the panel level where the user typically creates or edits data. As shown in FIG. 5, each panel object 506

represents an atomic step in task 500. Panels 504, including panel objects 506, create form elements that enable windows for data to be entered, displayed, or edited. A form element is an HTML construct that defines the fields that will be sent to a server when a user clicks a SUBMIT button. A panel object 506 is responsible for knowing where to go in the data objects to get its default data. This default data is used for describing the requirements for valid edits of the data and for describing where in the system to store valid results.

In the Office Action, the Examiner referred to pages 7-9 and 18-21 of Homestead as describing the “panel objects” feature. The Office Action asserts, according to Homestead, that a user “creates a new page from a template of data objects and goes on to open the Homestead online editor in a new panel to edit elements via the addition/deletion of elements on the user’s page. . . . Thus, Homestead discloses a sequence of screens / panels the user must go through to manipulate objects for the page creation process.” (Office Action, page 11, paragraph 2). The Appellants respectfully disagree.

Once again, the Examiner is attempting to equate web site building software using drag-and-drop Elements with unrelated functionality, having no basis for the comparison. The descriptions of web site customization on pages 7-9 and 18-21 of Homestead do not provide any disclosure or suggestion as to the nature of the various software objects which may be employed to implement the functionalities described. For the sake of argument, even if the Homestead online editor is presented in a “panel,” Homestead only enables association and incorporation of a predefined Element or application with the web site being created. There is certainly no disclosure or suggestion of actually “editing” those Elements, contrary to the assertion in the Office Action. One could argue that the web page being customized is edited, but the Elements themselves are not. They are simply selected, dragged and dropped. Frankly, Homestead describes nothing something similar to the claimed panel objects’

functionality: (1) being within a sequence through which data related to the multi-application program is entered and manipulated, and wherein (2) a panel object is aware of which data object from the plurality of data objects to access to retrieve existing data related to the multi-application program.

In Homestead, incorporation of a predefined Element with a web site being created does not involve “entering” data through a panel, where the data entered is related to the multi-application program. Assuming the Element constitutes such data, the Element already exists and is stored on the Element palette. Thus, the Element is not “entered” through any sequence of panels. In addition, in Homestead, any panel object would not be “aware of which data from the plurality of data objects to access . . .” If anything, Homestead is dependent upon user input to instruct the system which Element to access from the palette, because the user must select from the predefined Elements to customize the web page.

In Homestead, the general description of “Site Manager” functionality which allows the creation, editing, viewing, and deletion of web pages, without any evidence regarding the nature of the objects underlying this functionality is simply not sufficient to disclose each and every feature of claim 24, as required to anticipate the claimed invention. Consequently, the rejection of claim 24 should be reversed.

Claim 25 is dependent upon claim 24 and is, therefore, not subject to rejection under 35 U.S.C. § 102(a) for at least the same reasons as claim 24.

B. The Homestead Reference Fails to Support a Rejection of Claim 31 Under 35 U.S.C. § 103(a).

Claim 31 was rejected as obvious in view of Homestead. The Examiner took Official Notice as to the feature of “developing an online transaction system for processing online

orders, including establishing an online account and checkout process.”

Claim 31 is dependent upon claim 26 which recites that “the method of building the customized Web site includes a task-based approach to completing an activity, employs an underlying, extensible data model, and has a uniform user experience.” As described above, the Homestead reference provides no information regarding such a feature. Thus, Homestead fails to support a rejection of claims 26 and 31 for at least the same reasons as claim 1. Further, Appellants respectfully submit that because “employ[ing] an underlying, extensible data model” is neither anticipated by nor obvious in view of Homestead, it would not have been obvious to those of ordinary skill in the art to establish an online account and checkout process in developing an online transaction system for processing orders made through a Web site built using an underlying, extensible data model. Again, reliance on Homestead to support the 35 U.S.C. § 103(a) rejection is misplaced. This rejection should be reversed.

C. The Homestead and Norcott References Fail to Support a Rejection of Claim 18 Under 35 U.S.C. § 103(a).

Claim 18 was rejected as obvious in view of Homestead and Norcott. The Office Action stated that “Homestead fails to explicitly disclose the data schema being implemented as a relational database.” (Office Action, page 9, paragraph 36). Appellants agree with this assessment of Homestead.

Claim 18 is dependent upon claim 13 which recites “a data model for arranging and configuring application data, wherein the application data are one of a fixed attribute and an extended attribute.” As explained above with respect claim 1, the Homestead reference provides no information regarding extensible data models or application data in the form of extended attributes. Thus, Homestead fails to support a rejection of claims 13 and 18 for at

least the same reasons as claim 1. Norcott fails to cure this deficiency of Homestead even if Norcott could somehow be combined therewith. Norcott only describes database management techniques, and offers nothing pertaining to an “integrated software platform for creating a user application having a user experience” having extensible data models per claims 13 and 18. Accordingly, this rejection should also be reversed.

D. Conclusion

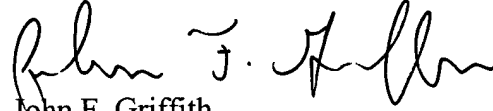
In view of the foregoing, all of the claim rejections under 35 U.S.C. §§ 102(a) and 103(a) based on the Homestead and Norcott references cannot stand for at least the reasons discussed. The Examiner’s reliance on Homestead and Norcott is misplaced, because the Examiner has failed to establish prima facie cases of anticipation and obviousness based on those references.

In view of the foregoing, Appellants respectfully request that the Board reverse the Examiner’s rejection of all pending claims. In addition, Appellants believe all claims now pending in this application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

Enclosed is the required fee for an appeal brief under 37 C.F.R. §1.17(c) in the amount of \$170, and a Petition for a One-Month Extension of Time.

Respectfully Submitted,

BEYER WEAVER & THOMAS, LLP

A handwritten signature in black ink, appearing to read "John F. Griffith", is written over the printed name.

John F. Griffith
Registration No. 44,137

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, CA 94704-0778
(510) 843-6200

VIII. CLAIMS APPENDIX

CLAIMS ON APPEAL

1. A software architecture for enabling multiple users to perform a plurality of tasks via a wide-area network, the software architecture comprising:
 - a plurality of applications;
 - a data schema for storing a plurality of data objects, the data schema having an underlying, extensible data model; and
 - an integrated platform for enabling each of the multiple users to perform at least one of the tasks by controlling interaction between two or more of the applications and the extensible data model.
2. A software architecture as recited in claim 1 wherein the integrated platform dynamically contextualizes each stage of a task with reference to a corresponding user.
3. A software architecture as recited in claim 2 wherein the integrated platform creates a contextual task list for the corresponding user.
4. A software architecture as recited in claim 1 wherein a data object is associated with a context information record that further describes a task in which the data object will be used.
5. A software architecture as recited in claim 1 wherein the platform allows a user and an application to extend the database in a user-specific way, thereby enabling the multiple users and the plurality of applications to use the database.

6. A software architecture as recited in claim 5 wherein the database can be extended dynamically without changing the underlying structure of the database.
7. A software architecture as recited in claim 1 wherein the extensible data model has extensible attributes that can be used to add a previously undefined data attribute and wherein the extensible, underlying data model provides a standard way of representing the previously undefined data attribute.
8. A software architecture as recited in claim 1 further comprising a user interface that is uniform across the plurality of applications.
9. A software architecture as recited in claim 1 wherein additional services can be added using the plurality of applications.
10. A software architecture as recited in claim 1 wherein the integrated platform is used to create and maintain an online business presence.
11. A software architecture as recited in claim 1 wherein the integrated platform is used to create and maintain a customer relationship management application.
12. A software architecture as recited in claim 1 wherein the architecture is a reactive architecture which supports a plurality of levels of task granularity and is dynamically aware of what information has been entered by a user.

13. An integrated software platform for creating a user application having a user experience comprising:

a data model for arranging and configuring application data, wherein the application data are one of a fixed attribute and an extended attribute;

a data logic component for operating on the data;

a back-end code layer for managing the user experience; and

a visual design component for implementing the user experience by presenting a user interface for entering data into a computer system.

14. An integrated software platform as recited in claim 13 further including an information architecture layer for modeling the user experience.

15. An integrated software platform as recited in claim 13 wherein the user application is a multiuser, online application.

16. An integrated software platform as recited in claim 13 wherein the user application is a customer relationship management application.

17. An integrated software platform as recited in claim 13 further including a data schema for storing data configured in the data model.

18. An integrated software platform as recited in claim 17 wherein the data schema is implemented as a relational database.

19. An integrated software platform as recited in claim 14 further including an interface layer for containing a plurality of HTML form elements.
20. An integrated software platform as recited in claim 19 wherein the interface layer and the back-end code layer translate instructions from the information architecture layer.
21. A task-based architecture for building a multi-user, online application by completing a plurality of tasks comprising:
- a data schema for storing data related to the online application;
 - an extensible data model for storing and sharing the data as the plurality of tasks is completed;
 - a plurality of tools;
 - a task viewer application for creating a user interface; and
 - a plurality of services for gathering and authoring the data.
22. A task-based architecture as recited in claim 21 wherein the plurality of tools includes a data extension framework for defining and extracting data.
23. A task-based architecture as recited in claim 21 wherein the plurality of tools includes a context management tool for determining a context in the user-oriented application development system.
24. A system for building a distributed, multi-application program comprising:

a plurality of tasks;
one or more sequences within a task;
a plurality of data objects; and
one or more panel objects within a sequence through which data related to the multi-application program is entered and manipulated, wherein a panel object is aware of which data object from the plurality of data objects to access to retrieve existing data related to the multi-application program.

25. A system as recited in claim 24 further comprising a plurality of model objects, a model object containing one or more data objects and a logic component for operating on the one or more data objects.

26. A method of building a customized Web site comprising:
creating and maintaining one or more Web pages;
developing a communication service with users accessing the Web site;
developing an online transaction system for processing online orders made through the Web site; and
creating a reporting service for generating reports relating to Web site activity, wherein the method of building the customized Web site includes a task-based approach to completing an activity, employs an underlying, extensible data model, and has a uniform user experience.

27. A method as recited in claim 26 further including creating and maintaining a catalog for describing one of one or more products and one or more services.

28. A method as recited in claim 26 wherein the task-based approach further includes completing one or more sequences, a sequence including one or more panels.
29. A method as recited in claim 26 wherein creating and maintaining one or more Web pages further includes controlling the appearance and content of a Web page.
30. A method as recited in claim 26 wherein developing a communication service with users further includes developing a user database according to user behavior patterns and preferences.
31. A method as recited in claim 26 wherein developing an online transaction system for processing online orders further includes establishing an online account and checkout process.